Learning Mixtures of Polynomials of Conditional Densities from Data

L. López-Cruz, Pedro; Nielsen, Thomas Dyhre; Bielza, Concha; Larrañga, Pedro

Published in:
Advances in Artificial Intelligence

DOI (link to publication from Publisher):
10.1007/978-3-642-40643-0_37

Publication date:
2013

Document Version
Early version, also known as pre-print

Learning Mixtures of Polynomials of Conditional Densities from Data

Pedro L. López-Cruz[1], Thomas D. Nielsen[2], Concha Bielza[1], and Pedro Larrañaga[1]

[1] Department of Artificial Intelligence, Universidad Politécnica de Madrid, Spain
pedro.lcruz@upm.es, (mcbielza, pedro.larranaga)@fi.upm.es
[2] Department of Computer Science, Aalborg University, Denmark
tdn@cs.aau.dk

Abstract. Mixtures of polynomials (MoPs) are a non-parametric density estimation technique for hybrid Bayesian networks with continuous and discrete variables. We propose two methods for learning MoP approximations of conditional densities from data. Both approaches are based on learning MoP approximations of the joint density and the marginal density of the conditioning variables, but they differ as to how the MoP approximation of the quotient of the two densities is found. We illustrate the methods using data sampled from a simple Gaussian Bayesian network. We study and compare the performance of these methods with the approach for learning mixtures of truncated basis functions from data.

Keywords: Hybrid Bayesian networks, conditional density estimation, mixtures of polynomials

1 Introduction

Mixtures of polynomials (MoPs) [1, 2], mixtures of truncated basis functions (MoTBFs) [3], and mixtures of truncated exponentials (MTEs) [4] have been proposed as density estimation techniques in hybrid Bayesian networks (BNs) including both continuous and discrete random variables. These classes of densities are closed under multiplication and marginalization, and they therefore support exact inference schemes based on the Shenoy-Shafer architecture. Also, the densities are flexible in the sense that they do not impose any structural constraints on the model, unlike, e.g., conditional linear Gaussian networks.

Only marginal and conditional MoTBFs appear during inference in hybrid BNs [5]. Learning MoP, MoTBF and MTE approximations of one-dimensional densities from data has been studied in [6, 7]. Learning conditional density approximations has, however, only been given limited attention [7, 8]. The main difficulty is that the classes of functions above are not closed under division. The general approach shared by existing methods for learning conditional densities is that the conditioning variables are discretized, and a one-dimensional approximation of the density of the conditional variable is found for each combination

of the (discretized) values of the conditioning variables. Thus, the estimation of a conditional density is equivalent to estimating a collection of marginal densities, where the correlation between the variable and the conditioning variables is captured by the discretization procedure.

In this paper, we present two new approaches, based on conditional sampling and interpolation, respectively, for learning MoP approximations of conditional densities from data. Our approach differs from previous methods in several ways. As opposed to [1–3], we learn conditional MoPs directly from data without any parametric assumptions. Also, we do not rely on a discretization of the conditioning variables to capture the correlation among the variables [7, 8]. On the other hand, our conditional MoPs are not proper conditional densities, hence posterior distributions established during inference have to be normalized so that they integrate to 1.

The paper is organized as follows. Section 2 briefly introduces MoPs and details the two new approaches for learning conditional MoPs. Experimental results and a comparison with MoTBFs are shown in Sect. 3. Section 4 ends with conclusions and outlines future work.

2 Learning Conditional Distributions

2.1 Mixtures of Polynomials

Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a multi-dimensional continuous random variable with probability density $f_{\mathbf{X}}(\mathbf{x})$. A MoP approximation of $f_{\mathbf{X}}(\mathbf{x})$ over a closed domain $\Omega_{\mathbf{X}} = [\epsilon_1, \xi_1] \times \cdots \times [\epsilon_n, \xi_n] \subset \mathbb{R}^n$ [1] is an L-piece d-degree piecewise function of the form

$$\varphi_{\mathbf{X}}(\mathbf{x}) = \begin{cases} pol_l(\mathbf{x}) & \text{for } \mathbf{x} \in A_l,\ l = 1, \ldots, L, \\ 0 & \text{otherwise,} \end{cases}$$

where $pol_l(\mathbf{x})$ is a multivariate polynomial function with degree d (and order $r = d + 1$) and $A_1, \ldots, A_L$ are disjoint hyperrectangles in $\Omega_{\mathbf{X}}$, which do not depend on $\mathbf{x}$, with $\Omega_{\mathbf{X}} = \cup_{l=1}^{L} A_l$, $A_i \cap A_j = \emptyset, i \neq j$.

Following the terminology used for BNs, we consider the conditional random variable X as the child variable and the vector of conditioning random variables $\mathbf{Y} = (Y_1, \ldots, Y_n)$ as the parent variables. Given a sample $\mathcal{D}_{X,\mathbf{Y}} = \{(x_i, \mathbf{y}_i)\}, i = 1, \ldots, N$, from the joint density of $(X, \mathbf{Y})$, the aim is to learn a MoP approximation $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$ of the conditional density $f_{X|\mathbf{Y}}(x|\mathbf{y})$ of $X|\mathbf{Y}$ from $\mathcal{D}_{X,\mathbf{Y}}$.

2.2 Learning Conditional MoPs Using Sampling

The proposed method is based on first obtaining a sample from the conditional density of $X|\mathbf{Y}$ and then learning a conditional MoP density from the sampled values. Algorithm 1 shows the main steps of the procedure. First, we find a MoP representation of the joint density $\varphi_{X,\mathbf{Y}}(x, \mathbf{y})$ (step 1) using the B-spline interpolation approach proposed in [6]. Second, we obtain a MoP of the marginal

density of the parents $\varphi_{\mathbf{Y}}(\mathbf{y})$ by marginalization (step 2). Next, we use a sampling algorithm to obtain a sample $\mathcal{D}_{X|\mathbf{Y}}$ from the conditional density of $X|\mathbf{Y}$ (step 3), where the conditional density values are obtained by evaluating the quotient $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$. More specifically, we have used a standard Metropolis-Hastings sampler for the reported experimental results. For the sampling process we generate uniformly distributed values over $\Omega_{\mathbf{Y}}$ for the parent variables $\mathbf{Y}$, whereas the proposed distribution for the child variable is a linear Gaussian distribution $\mathcal{N}(\boldsymbol{\beta}^T\mathbf{y},\sigma^2)$, where $\boldsymbol{\beta}$ is an n-dimensional vector with all components equal to $1/n$. We used $\sigma^2 = 0.5$ in our experiments. Next, we find an (unnormalized) MoP approximation of the conditional density $X|\mathbf{Y}$ from $\mathcal{D}_{X|\mathbf{Y}}$ (step 4). Finally, we apply the partial normalization procedure proposed in [1] to obtain a MoP approximation $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$ of the conditional density (steps 5 and 6). The complexity of the algorithm is dominated by the complexity of the learning algorithm in [6].

This method has some interesting properties. The B-spline interpolation algorithm for learning MoPs in [6] guarantees that the approximations are continuous, non-negative and integrate to one. Therefore, the conditional MoPs obtained with Algorithm 1 are also continuous and non-negative. Continuity is not required for inference in BNs, but it usually is a desirable property, e.g., for visualization purposes. The algorithm provides maximum likelihood estimators of the mixing coefficients of the linear combination of B-splines when learning MoPs of the joint density ($\varphi_{X,\mathbf{Y}}(x,\mathbf{y})$) and the marginal density $\varphi_{\mathbf{y}}(\mathbf{y})$, hence the quotient $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$ corresponds to a maximum likelihood model of the conditional distribution. It should be noted, though, that this property is not shared by the final learned model as the partial normalization (steps 5 and 6) does not ensure that the learned MoP is a proper conditional density. Therefore, the MoP approximations of the posterior densities should be normalized to integrate to 1.

Algorithm 1.
Inputs:

- $\mathcal{D}_{X,\mathbf{Y}}$*: A training dataset* $\mathcal{D}_{X,\mathbf{Y}} = \{(x_i,\mathbf{y}_i)\}, i = 1,\ldots,N$
- r*: The order of the MoP*
- L*: The number of pieces of the MoP*

Output: $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$*. The MoP approximation of the density of* $X|\mathbf{Y}$
Steps:

1. *Learn a MoP* $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})$ *of the joint density of* $(X,\mathbf{Y})$ *from the dataset* $\mathcal{D}_{X,\mathbf{Y}}$ *using polynomials with order* r *and* L *pieces* [6].
2. *Marginalize out* X *from* $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})$ *to yield a MoP* $\varphi_{\mathbf{Y}}(\mathbf{y})$ *of the marginal density of the parent variables* $\mathbf{Y}$*:* $\varphi_{\mathbf{Y}}(\mathbf{y}) = \int_{\Omega_X} \varphi_{X,\mathbf{Y}}(x,\mathbf{y})dx$.
3. *Use a Metropolis-Hastings algorithm to yield a sample* $\mathcal{D}_{X|\mathbf{Y}}$ *with* M *observations from the conditional density* $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$.
4. *Learn an unnormalized conditional MoP* $\varphi^{(u)}_{X|\mathbf{Y}}(x|\mathbf{y})$ *from* $\mathcal{D}_{X|\mathbf{Y}}$ *using polynomials with order* r *and* L *pieces* [6].

5. *Compute the partial normalization constant:*

$$c = \int_{\Omega_X} \int_{\Omega_{\mathbf{Y}}} \varphi_{\mathbf{Y}}(\mathbf{y}) \varphi^{(u)}_{X|\mathbf{Y}}(x|\mathbf{y}) d\mathbf{y} dx \ .$$

6. *Find the partially normalized MoP of the conditional density:*

$$\varphi_{X|\mathbf{Y}}(x|\mathbf{y}) = \frac{1}{c} \varphi^{(u)}_{X|\mathbf{Y}}(x|\mathbf{y}) \ .$$

We show an example with two variables X and Y. We sampled a training dataset $\mathcal{D}_{X,Y}$ with $N = 5000$ observations from the two-dimensional Gaussian density $(X, Y) \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}\right)$. This two-dimensional density corresponds to a Gaussian BN, where $Y \sim \mathcal{N}(0, 1)$ and $X|Y \sim \mathcal{N}(y, 1)$. Next, we applied Algorithm 1 to learn the MoP approximation of the conditional density of $X|Y$. The domain of the approximation was set to $\Omega_{X,Y} = [-3, 3] \times [-2, 2]$, which includes 0.9331 of the total Gaussian density mass. Note that $\sigma^2_Y = 1$ is smaller than $\sigma^2_X = 2$, thus the domain $\Omega_Y = [-2, 2]$ is smaller than Ω_X. We used the BIC score to greedily find the number of pieces L and the order r of the MoP. The conditional MoP learned with Algorithm 1 is shown in Fig. 1(a). The conditional MoP had $L = 16$ pieces and order $r = 2$, i.e., 64 polynomial coefficients. The true conditional density of $X|Y$ is the linear Gaussian density $\mathcal{N}(y, 1)$ shown in Fig. 1(b). We can see that the conditional MoP in Fig. 1(a) is continuous and close to the true conditional density. We observe high peaks at the "corners" of the domain $\Omega_{X,Y}$. These are due to numerical instabilities when evaluating the quotient $\varphi_{X,Y}(x, y)/\varphi_Y(y)$, caused by both the joint and the marginal MoPs yielding small values (close to zero) at the limits of the approximation domain.

Next, we performed inference based on the conditional MoP learned with Algorithm 1. Figures 1(c), (d) and (e) show the MoPs (solid) and true (dashed) posterior densities for Y given three different values for X. The three values correspond to the percentiles 10, 50 and 90 of $X \sim \mathcal{N}(0, 2)$. Both the MoPs and the true posterior densities shown in Figs. 1(c), (d) and (e) were normalized in the domain Ω_Y so that they integrate to one. We can see that the MoPs of the posterior densities are also continuous and close to the true posterior densities; Kullback-Leibler divergence values are reported in Sect. 3.

2.3 Learning Conditional MoPs Using Interpolation

The preliminary empirical results output by Algorithm 1 show that the sampling approach can produce good approximations. However, it is difficult to control or guarantee the quality of the approximation due to the partial normalization.

This shortcoming has motivated an alternative method for learning a MoP approximation of a conditional probability density for $X|\mathbf{Y}$. The main steps of the procedure are summarized in Algorithm 2. First, we find MoP approximations of both the joint density of $(X, \mathbf{Y})$ and the marginal density of $\mathbf{Y}$ in the

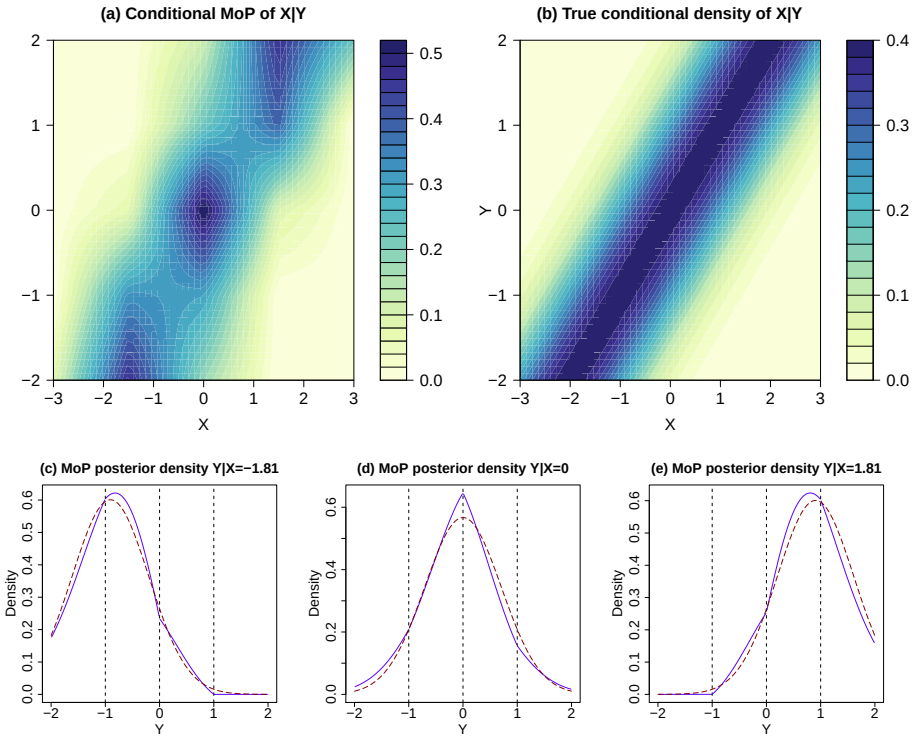

Fig. 1: (a) Conditional MoP of $X|Y$ learned with Algorithm 1. (b) True conditional density of $X|Y \sim \mathcal{N}(y, 1)$. (c,d,e) MoP approximations (solid) and true posterior densities (dashed) of $Y|X$ for three values of X.

same way as in Algorithm 1 (steps 1 and 2). Next, we build the conditional MoP $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$ by finding, for each piece $pol_l(x, \mathbf{y})$ defined in the hyperrectangle A_l, a multidimensional interpolation polynomial of the function given by the quotient of the joint and the marginal densities $\varphi_{X,\mathbf{Y}}(x, \mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$.

Algorithm 2.
Inputs:

- $\mathcal{D}_{X,\mathbf{Y}}$*: A training dataset* $\mathcal{D}_{X,\mathbf{Y}} = \{(x_i, \mathbf{y}_i)\}, i = 1, \ldots, N$
- r*: The order of the MoP*
- L*: The number of pieces of the MoP*

Output: $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$*. The MoP approximation of the density of* $X|\mathbf{Y}$
Steps:

1. *Learn a MoP* $\varphi_{X,\mathbf{Y}}(x, \mathbf{y})$ *of the joint density of the variables* X *and* $\mathbf{Y}$ *from the dataset* $\mathcal{D}_{X,\mathbf{Y}}$ [6].
2. *Marginalize out* X *from* $\varphi_{X,\mathbf{Y}}(x, \mathbf{y})$ *to yield a MoP* $\varphi_{\mathbf{Y}}(\mathbf{y})$ *of the marginal density of the parent variables* $\mathbf{Y}$*:* $\varphi_{\mathbf{Y}}(\mathbf{y}) = \int_{\Omega_X} \varphi_{X,\mathbf{Y}}(x, \mathbf{y})dx$.

3. *For piece* $pol_l(x, \mathbf{y})$*, defined in* A_l*,* $l = 1, \ldots, L$*, in the conditional MoP* $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$*:*
   Find a multi-dimensional polynomial approximation of function $g(x, \mathbf{y}) = \varphi_{X,\mathbf{Y}}(x, \mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$ *using an interpolation method.*

We consider two multidimensional interpolation methods, which can be used to obtain the polynomials of the pieces $pol_l(x, \mathbf{y})$ in step 3 of Algorithm 2:

- The multidimensional Taylor series expansion (TSE) for a point yields a polynomial approximation of any differentiable function g. The quotient of any two functions is differentiable as long as the two functions are also differentiable. In our scenario, polynomials are differentiable functions and, thus, we can compute the TSE of the quotient of two polynomials. Consequently, we can use multidimensional TSEs to find a polynomial approximation of $g(x, \mathbf{y}) = \varphi_{X,\mathbf{Y}}(x, \mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$ for each piece $pol_l(x, \mathbf{y})$. We computed these TSEs of $g(x, \mathbf{y})$ for the midpoint of the hyperrectangle A_l.
- Lagrange interpolation (LI) finds a polynomial approximation of any function g. Before finding the LI polynomial, we need to evaluate function g on a set of interpolation points. In the one-dimensional scenario, Chebyshev points are frequently used as interpolation points [9]. However, multidimensional LI is not a trivial task because it is difficult to find good interpolation points in a multidimensional space. Some researchers have recently addressed the two-dimensional scenario [9, 10]. To find a conditional MoP using LI, we first find and evaluate the conditional density function $g(x, \mathbf{y}) = \varphi_{X,\mathbf{Y}}(x, \mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$ on the set of interpolation points in A_l. Next, we compute the polynomial $pol_l(x, \mathbf{y})$ for the piece as the LI polynomial over the interpolation points defined in A_l. Note that other approaches, e.g., kernel-based conditional estimation methods, can also be used to evaluate the conditional density $g(x, \mathbf{y})$ on the set of interpolation points.

Compared with Algorithm 1, there are some apparent (dis)advantages. First, the conditional MoPs produced by Algorithm 2 are not necessarily continuous. Second, interpolation methods cannot in general ensure non-negativity, although LI can be used to ensure it by increasing the order of the polynomials. On the other hand, the learning method in Algorithm 2 does not need a partial normalization step. Thus, if the polynomial approximations are close to the conditional density $\varphi_{X,\mathbf{Y}}(x, \mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$, then the conditional MoP using these polynomial interpolations is expected to be close to normalized. As a result, we can more directly control the quality of the approximation by varying the degree of the polynomials and the number of hyperrectangles.

We applied Algorithm 2 to the example in Fig. 1. We used the two-dimensional LI method over the Padua points in [10] to compute the polynomials $pol_l(x, \mathbf{y})$ of the conditional MoP, see Fig. 2(a). The conditional MoP with the highest BIC score had $L = 16$ pieces and order $r = 3$, i.e., 144 polynomial coefficients. We observe that the conditional MoP in Fig. 2(a) is not continuous. Also, the MoPs of the posterior density in Figs. 2(c), (d) and (e) are not continuous either; Kullback-Leibler divergence values are reported in Sect. 3.

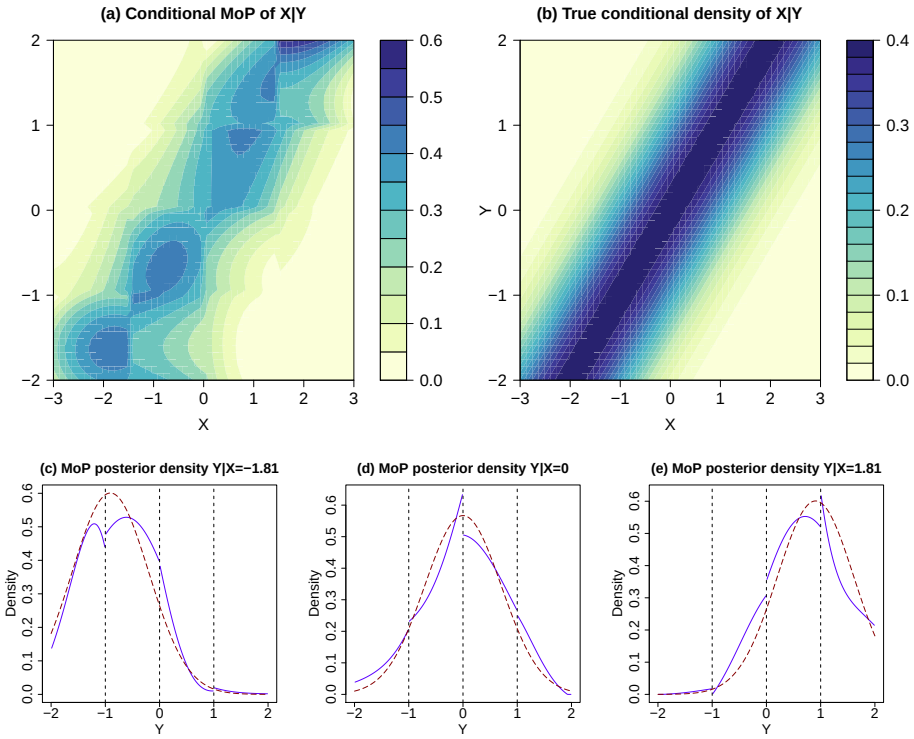

Fig. 2: (a) Conditional MoP of $X|Y$ learned with Algorithm 2. (b) True conditional density of $X|Y \sim \mathcal{N}(y, 1)$. (c,d,e) MoP approximations (solid) and true posterior densities (dashed) of $Y|X$ for three values of X.

3 A Comparison with MoTBFs

In this section, we compare the approaches proposed in this paper with the method proposed in [7] for learning conditional MoTBFs from data. Figure 3 shows the MoTBFs of the conditional (a) and the posterior (c,d,e) densities approximated using the data in Figs. 1 and 2. The conditional MoTBF had $L = 6$ pieces and each piece defined a MoP with at most six parameters. MoTBF approximations of conditional densities are obtained by discretizing the parent variables and fitting a one-dimensional MoTBF for each combination of the discrete values of the parents. Compared with the two learning methods proposed in Algorithms 1 and 2, the method in [7] captures the correlation between the parent variables and the child variable through the discretization instead of directly in the functional polynomial expressions.

If there is a weak correlation between the child and parent variables, then the conditional MoTBF approach is expected to yield approximations with few pieces. On the other hand, as the variables become more strongly correlated, ad-

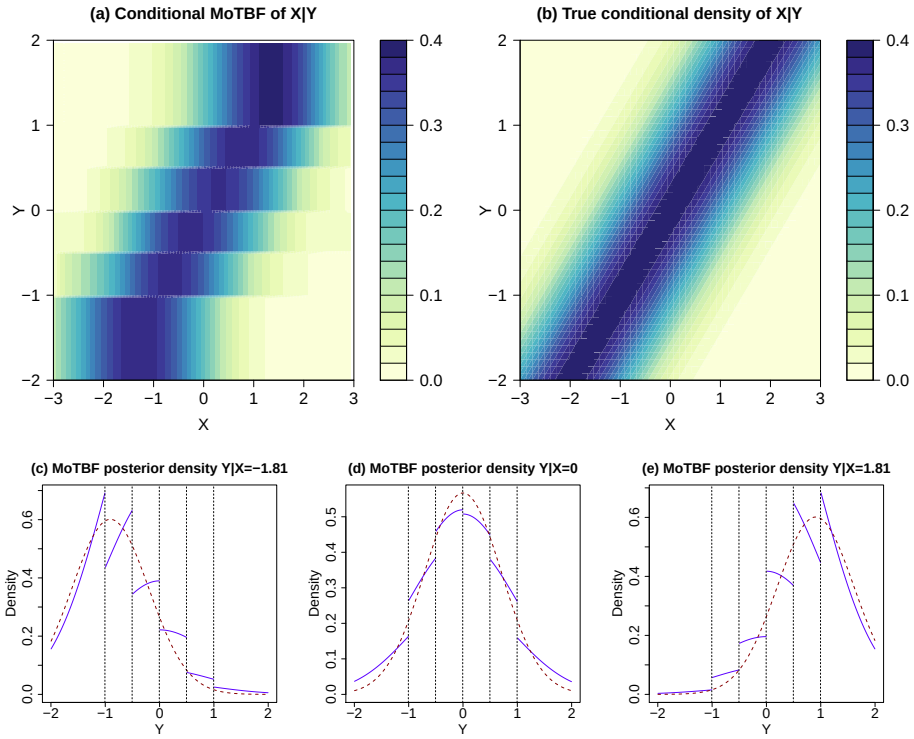

Fig. 3: (a) Conditional MoTBF of $X|Y$ learned with the approach in [7]. (b) True conditional density of $X|Y \sim \mathcal{N}(y, 1)$. (c,d,e) MoTBF approximations (solid) and true posterior densities (dashed) of $Y|X$ for three values of X.

ditional subintervals will be introduced by the learning algorithm. The MoTBF learning algorithm does not rely on a discretization of the child variable, but it rather approximates the density using a higher-order polynomial/exponential function. In contrast, Algorithms 1 and 2 yield conditional MoPs with more pieces because the domain of approximation $\Omega_{X,\mathbf{Y}}$ is split into hyperrectangles in all the dimensions. However, with the finer-grained division of the domain into hyperrectangles, the polynomial functions of the conditional MoPs will usually have a low order.

We empirically compared the results of Algorithm 1, Algorithm 2 (using both TSE and LI) and the method proposed in [7]. We sampled ten datasets for each sample size ($N = 25, 500, 2500, 5000$) from the Gaussian BN, where $Y \sim \mathcal{N}(0, 1)$ and $X|Y \sim \mathcal{N}(y, 1)$. We used Algorithms 1 and 2 as part of a greedy search procedure. We started by considering one interval for each dimension ($L = 1$) and order $r = 2$ (linear polynomials). Then, we increased either the number of intervals to 2 ($L = 4$) or the order of the polynomials to $r = 3$. Finally, we chose the MoP with the highest BIC score out of the two MoPs (increasing

Table 1: Mean Kullback-Leibler divergences between the MoP approximations and the true posterior densities for ten datasets sampled from the BN, where $Y \sim \mathcal{N}(0,1)$ and $X|Y \sim \mathcal{N}(y,1)$. The best results for each sample size are highlighted in bold. Statistically significant differences at $\alpha = 0.05$ are shown with symbols $*, \dagger, \ddagger, \star$.

N	$Y\|X=x$	Alg. 1 ($*$)		Alg. 2 TSE ($\dagger$)		Alg. 2 LI ($\ddagger$)		MoTBF ($\star$)	
25	X =-1.81	0.5032	$\dagger\star$	0.7297		**0.3487**	$*\dagger\star$	0.7084	$\dagger$
	X =0.00	0.0746	$\ddagger\star$	**0.0745**	$*\ddagger\star$	0.1510		0.0939	$\ddagger$
	X =1.81	**0.4952**	$\dagger\ddagger\star$	0.7297	$\ddagger$	1.4582		0.7084	$\dagger\ddagger$
500	X =-1.81	0.4194		0.2321	$*\ddagger$	0.3161	$*$	**0.2191**	$*\ddagger$
	X =0.00	**0.0239**	$\dagger\ddagger\star$	0.0646	$\star$	0.0453	$\dagger\star$	0.0950	
	X =1.81	0.4141		0.2311	$*\ddagger$	0.3701	$*$	**0.2170**	$*\ddagger$
2500	X =-1.81	0.1045		0.0850		0.1128		**0.0728**	$*\ddagger$
	X =0.00	0.0387		0.0441		**0.0097**	$*\dagger\star$	0.0272	$*\dagger$
	X =1.81	0.0984		0.0978		0.1041		**0.0695**	$*\ddagger$
5000	X =-1.81	0.0575		0.0413		0.0341	$*$	**0.0308**	$*$
	X =0.00	**0.0196**		0.0262		0.0221		0.0210	
	X =1.81	0.0556		0.0425		0.0383		**0.0322**	$*$

either L or r) and iterated until there was no further increase in the BIC score. Table 1 shows the mean Kullback-Leibler divergences between the MoPs and the true posterior densities $Y|X$ for three values of X in the ten repetitions. We applied a paired Wilcoxon signed-rank test and report statistically significant differences at a significance level $\alpha = 0.05$. The null hypothesis is that the two methods perform similarly. The alternative hypothesis is that the algorithm in the column outperforms the algorithm shown with a symbol: $*$ for Alg. 1, $\dagger$ for Alg. 2 with TSE, $\ddagger$ for Alg. 2 with LI, and $\star$ for conditional MoTBFs. For instance, a $\star$ in the column corresponding to Alg. 1 in Table 1 shows that Alg. 1 significantly outperformed MoTBFs for a given value of N and X. Algorithms 1 and 2 yielded competitive results against conditional MoTBFs.

4 Conclusion

We have presented two methods for learning MoP approximations of the conditional density of $X|\mathbf{Y}$ from data. Both methods are based on finding MoP approximations of the joint density $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})$ and the marginal density of the parents $\varphi_{\mathbf{Y}}(\mathbf{y})$. Thus, the first method obtains a sample from the conditional density $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$ using a Metropolis-Hastings algorithm, from which it learns the conditional MoP $\varphi_{X|\mathbf{Y}}(x|\mathbf{y})$. The second method obtains a MoP of the conditional density $\varphi_{X,\mathbf{Y}}(x,\mathbf{y})/\varphi_{\mathbf{Y}}(\mathbf{y})$ using a multidimensional interpolation technique. Multidimensional TSE and LI were considered and evaluated. The approaches were empirically studied and compared with MoTBFs using a dataset sampled from a Gaussian BN. As opposed to previous research on approximating conditional densities, the proposed approaches rely only on data

without assuming any prior knowledge on the generating parametric density. Also, continuous parents do not need to be discretized.

In this paper, the same number of intervals were used for learning the MoPs of the joint and the conditional densities. Also, equal-width intervals $[\epsilon_i, \xi_i]$ are considered in each dimension, and the hyperrectangles A_l have the same size. In the future, we intend to study how to automatically find appropriate values for the order r, the number of pieces L, and the limits $[\epsilon_i, \xi_i]$ of the hyperrectangles defining each one of the MoPs. This should reduce the number of pieces required to find good MoP approximations. We also intend to use these approaches in more complex BNs. This involves considering other problems, e.g., BN structure learning. Finally, we intend to thoroughly compare these methods with MTE and MoTBF approaches.

Acknowledgments. This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through Cajal Blue Brain (C080020-09) and TIN2010-20900-C04-04 projects. PLLC is supported by a Fellowship (FPU AP2009-1772) from the Spanish Ministry of Education, Culture and Sport.

References

1. Shenoy, P.P., West, J.C.: Inference in hybrid Bayesian networks using mixtures of polynomials. Int. J. Approx. Reason. 52, 641–657 (2011)
2. Shenoy, P.P.: Two issues in using mixtures of polynomials for inference in hybrid Bayesian networks. Int. J. Approx. Reason. 53, 847–866 (2012)
3. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Mixtures of truncated basis functions. Int. J. Approx. Reason. 53, 212–227 (2012)
4. Moral, S., Rumí, R., Salmerón, A.: Mixtures of truncated exponentials in hybrid Bayesian networks. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS, vol. 2143, pp. 145–167. Springer, Heildelberg (2001)
5. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Inference in hybrid Bayesian networks with mixtures of truncated basis functions. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models, pp. 163–170 (2012)
6. López-Cruz, P.L., Bielza, C., Larrañaga, P.: Learning mixtures of polynomials from data using B-spline interpolation. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models, pp. 211–218 (2012)
7. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Learning mixtures of truncated basis functions from data. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models, pp. 163–170 (2012)
8. Langseth, H., Nielsen, T.D., Rumí, R., Salmerón, A.: Maximum likelihood learning of conditional MTE distributions. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS, vol. 5590, pp. 240–251. Springer, Heildelberg (2009)
9. Harris, L.A.: Bivariate Lagrange interpolation at the Chebyshev nodes. Proc. Amer. Math. Soc. 138, 4447–4453 (2010)
10. Caliari, M., De Marchi, S., Sommariva, A., Vianello, M.: Padua2DM: Fast interpolation and cubature at the Padua points in Matlab/Octave. Numer. Algorithms 56, 45–60 (2011)